Newark Medical Associates

Application Report

Luis Serpas

Table of Content

1. Introduction

This report is composed of two sections. The first section is called the Technical Report which describes the individual components of the Newark Medical Associates Database and Web Front End. They components are a series of php and html files organized into a folder hierarchy to prompt ease of modification and navigation.

The second section will guide a typical end user through the menu options and navigation of the applications front end.

2. Clients

The clients of this manual include the developers who will have access to the code with the intent to alter the front end, end users who will the NMA application only through the forms and buttons provided and to any user who wishes to create an instance of the exact application using the assets provided.

3. Necessary Configuration

Local apache server should be set up to run the application. MAMP could be used for MAC OS or XAMPP in Windows. The MySQL server should also be set up in order to run the database MySQL but this included with the XAMPP software.

The apache server is a directory located in the htdocs directory within the users program files in their c drive under XAMPP. The user simply needs to copy the entire folder NMA folder and place it into htdocs. Access to the website by using a browser to reach http://localhost/nma/home.php.

4. Components and their Assets

****All assets are of type php unless stated otherwise.**

Component	Assets		
Database Input	(Located in PHP folder)		
	account	addNurseSkill	deleteEmployee
	addAdmit	addOwner	deleteNurse
	addAllergy	addPatient	deletePhysician
	addClinic	addPatientAllergy	deleteSchedule
	addClinicBed	addPhysician	deleteStaff
	addClinicEmp	addPrescription	deleteSurgeon
	addConsult	addShift	getSafeData
	addCorporation	addShiftSchedule	scheduleSurgery
	addDiagnosis	addSurgeon	updateNurse
	addEmployee	addSurgery	updatePhysician
	addIllness	addSurgeryReq	connect.
	addMedical	addSurgerySkill	getSafeData.php
	addMedication	addSurgeryType	Login.php
	addMedReact	deleteAdmision	addNurse

Front End Display and Form	(Located in root NMA folder)		
	addMisc	viewAllergy	viewPatientAllergy
	addPatient	viewClinic	viewPhysician
	addScheduleShift	viewClinicBed	viewPrescription
	addShiftSchedule	viewClinicEmp	viewReactWith
	addStaff	viewConsult	viewSchedule
	footbar.html	viewCorporation	viewScheduledSurgery
	home	viewDiagnosis	viewSurgeon
	inPatientManage	viewEmployee	viewSurgery
	miscManage	viewIllness	viewSurgeryReq
	navBar.html	viewMedical	viewSurgerySkill
	schedule	viewMedication	viewSurgeryType
	staffManage	viewNurse	viewOnwer
	updateNurse	viewNurseSkill	viewPatient
	updatePhysician	viewAdmitted	
Javascript Menu PopUp	(Located In assets folder)		
	nmaButton.js		
Front End Theme and Style	(Located In js folder)		
	nmaStyle.css		

5. Technical Report

Section 4 listed all the assets that make up the Newark Medical Associates Application. Further it breaks it into components that share some similarities in functionality. This similarity is also translated to the code which would be nearly identical copies were it not for their access to different tables and different sql statements.

In this section I attempt to provide a better understanding of the assets that make up the individual components of the NMA application. Beginning with the Database input.

Database Input

The assets under this component reside in the PHP folder of the NMA application and their purpose is to be called through a post action that occurs in the forms located in the applications front end. These assets include addPrescription, addMedicine, addPatient.

The typical code within each of this files is similar to the following:

**This is sample code found in the addEmployee php file

All the php files and indeed any section of code that makes use of php includes the first three lines that import the account, connect, and getSafeData files that are stored in the php folder. Their purpose of the account file, for instance, is to specify the account and password used to login to the mySql backend.

```
<?php $hostname = "localhost";
$username = "root";
$password= "toor";
$project = "nma";?>
```

The connect file then uses the account in order to login to the backend.

```
 //account connection
 $dbh = mysql_connect($hostname, $username, $password) or die ("Error");
 //database selection
 mysql_select_db($project);
```

The getSafeData file which is used throughout all the input files makes sure that any input is sanitized to avoid sql injection.

```
 include ("account.php");
 include ("connect.php");
 include ("getSafeData.php");
 $id = getSafeData($_POST["empid"]);
 $name = getSafeData($_POST["name"]);
…
//all these variables will get the information that is submitted into the forms
$s = "insert into employee(empid, name, salary, gender, address, type, ssn, phone)
 values ('$id',    '$name', '$salary', '$gender', '$address', '$type', '$ssn', '$phone')" ;

if(!empty($id) && (((int)$salary)>25000) &&(((int)$salary)<300000))
 {
 mysql_query ( $s ) or print (mysql_error());
…}
```

Please note that the if statement found here shows an example of a business rule that I chose to implement here in the php file rather than the database. Although I had attempted to use triggers or an enumerated column this method proved to be simple and easily mimicked throughout the other php files where other business rules were implanted another case such as this is the Risk of Heart condition business rule…

```
$riskOf = ($hdl+$ldl+((1/5)*$tri))/$hdl;
$risk = '';
if($riskOf<4){risk ='N';}
if((4<$riskOf)&&($riskOf<5)) {$risk ='L';}
if(5<$riskOf) {$risk ='M';}
```

<u>Front End Display and Forms</u>

The php files located in the root of the NMA folder are tasked with providing for the user the mechanism by which they can input data into the backend database and also retrieve, view, delete, and update certain table’s date such as a patient’s assigned nurse.

Typically files located here are following the naming convention of view[Table], for instance, viewNurse, viewEmployee, viewAllergy, viewIllness to name a few.

These files are all of the type php except for a few such as the navbar.html or footbar.html though renaming them wouldn’t change their functionality which is to provide the links for the user access the different front end sections.

The front end sections are the following and they are listed in the applications Navigation Bar they accessible from anywhere on the site.

Section	Form Available		
Add Patient	(addPatient)		
	Add Patient	Add Diagnosis	Add Allergy
	Add Consult	Add Prescription	
	Add Surgery	Add Medical	
View Patient	(patientManage)		
	Patients	Diagnosis	Patient Allergy
	Consultations	Prescription	
	Surgeries	Medical Data	
Add Staff	(addStaff)		
	Add Employee	Add Physician	Add Clinic Emp
	Add Nurse	Add Surgeon	
	Add Nurse Skill	Add Owner	
View Staff	(staffManage)		
	Employees	Physicians	Clinic Emp
	Nurses	Surgeons	
	Nurse Skills	Owners	

In Patient	(inPatientManage)		
	Add Bed	Admit To	Update Physician
	Clinic Beds	Admitted	Update Nurse
Add Misc.	(addMisc)		
	Add Allergy	Add Surgery Skill	Add Medication
	Add illness	Add Corporation	Add Med Reaction
	Add Surgery Type	Add Clinic	Add Surgery Require
View Misc.	(miscManage)		
	Allergy	Surgery Skills	Medications
	Illness	Corporations	Med Reactions
	Surgery Types	Clinics	Surgery Req Skills.
Schedule	(schedule)		
	Add Shift	View Schedule	Schedule
	Schedule Shift	Schedule Surgery	

This table breaks up the applications front end home page into the different links that allow the user to provide input for new employees, new nurses, new patients; it also provides for the user the ability to view table data such as view Nurse, and view Patients.

The front end divides the different functionality input / output into separate pages so that the user can input data for any given entity from the same menu option or vice a versa view any information about a given entity. This design feature is to simplify the page layout into obvious and specific functions that directly tie into the folder and asset hierarchy. This allows a developer to easily navigate to and update a piece of code from any asset that makes up the application easily and in exactly the same way for the different entities.

Type 1:

There are three basic front end php files. The first and the simplest type only provide the user with a list of form options through which they can input data. For example the php file for addPatient.php contains the following code.

```
<div id="selection" class="height">

<h2>Select Form</h2>

<table id="table" border ="1">

<tr><th>

<input class="button" type="button" value="Add Allergy"
onclick="Display('allergyForm')" /></th></tr></table></div>
```

A typical front end php file contains two parts; the first part is the div which shows the users the different forms that they can pick to view.

Upon loading the page all forms are rendered hidden. The way to make the forms visible is through the javascript file nmaButton. This javascript file is in charge of displaying the form the user wants to view based on which button he pressed.

The file is also charge of hiding any forms that had been previously viewed.

Html is not a dynamic language and typically the content is static, the javascript overcomes this and gives the page a bit more fluidity.

The second portion of a typical front end php file is the form that is displayed to the user and through which the user is able to post data the php files that are charge of directly accessing the backend using sql.

This form displays the data that makes up a typical allergy entity including the allergy code and description.

The addpatientAllergy php file is used to take data input through this form and inserts it into the database. This file is called upon the user pressing the submit button.

```
<form action="php/addPatientAllergy.php" method="post" id="allergyForm"
style="display:none">

<h2>New Patient Allergy</h2>

<table    border ="1">

<tr><th>Patient ID</th>td>input type="text" name="id" maxlength="5" /></td></tr>

<tr><th>Allergy Code</th>td><input type="text" name="allergy" maxlength="5"
/></td>tr>

<tr><th colspan="2"><input class="button" type="submit" value="Submit"/>

</th></tr></table></form>
```

Type 2 and 3:

The next type of php file that makes up the front end of my NMA application simply provides for the user a table populated with data from the backend, the table is further styled using the nmaStyle.css file located in the assets folder.

This is the typical code for displaying a table using php, sql, and css.

```
<?php

$selectAll=("select *from nurse, employee where nurse.nurseIdCode =
employee.empId;");

$result = mysql_query($selectAll) or die();

echo "<table id='phpTable'>";

echo "<tr><th>Actions</th>

<th style='width: 150px;'>ID</th>

<th style='width: 150px;'>Name</th>

<th style='width: 150px;'>Surgery Type</th>

<th style='width: 150px;'>Grade</th>

<th style='width: 150px;'>Years</th></tr>";

while($row = mysql_fetch_array($result))

{$code=$row['nurseIdCode'];

$name=$row['name'];

$surgery=$row['surgeryTypeCode'];

$grade=$row['grade'];

$years=$row['years'];
```

```
echo "<tr><td>

<a href='home.php'>home</a>

/<a href='php/deleteNurse.php?code=".$code."'>Delete</a></td></td>

<td>

<input class='noUpdate' readonly='readonly'    type='text' name='code' value='".$code."'
/></td>

<td>

<input class='noUpdate' readonly='readonly'    type='text' name='name'
value='".$name."' /></td>

<td>

<input class='noUpdate' readonly='readonly'    type='text' name='surgery'
value='".$surgery."' />

</td>

<td>

<input class='noUpdate' readonly='readonly'    type='text' name='grade'
value='".$grade."' />

</td>

<td>

<input class='noUpdate' readonly='readonly'    type='text' name='years'
value='".$years."' />

</td>
```

```
</tr>";

} // End our while loop

echo "</table>"; ?>
```

This piece of code displays the last two types of php files that are used to display data for the user and also allow the user to update the data directly.

In this case this table is displayed within another html file thereby inheriting all the stylistic effects already present in the original html file. This table merely need identify itself with an existing class element found in the nmaStyle.css file in order to be stylized according to the applications theme.

Furthermore the table allows a user to directly alter the data located in the table by placing the created table within a form whose submit button is placed in the column of the table.

This form, unlike the others previously used, sends data to the php files using a get method. The reason is this allowed me to directly name the variable I would be sending to the php file. In this case I send the variable code which is the Nurses User Id.

The id is sent to a php filed called deleteNurse and this uses the code to delete the entity that whose id matches it.

Another similar cases updates the information rather than the deletes it. The two sql statements are listed next:

In order to delete:

```
$code = $_GET['code'];

mysql_query("delete from nurse where nurseIdCode='$code'") or die(mysql_error());
```

Using an sql set in order to alter data.

```
$pnum = getSafeData($_POST["id"]);

$nurse = getSafeData($_POST["nurse"]);

$s = "update patient set attendingNurseId='$nurse' where pnum = '$pnum'";

mysql_query ( $s ) or print (mysql_error());
```

<u>Javascript Menu PopUp</u>

This script simply works like a switch and turns table visibility off, if already on, or on if already off. The form displayed or hidden depends on which form the user wants to see.

The id of which is sent to the script via a variable 'id' to the js function Display(id).

```
//hidding previous form

document.getElementById(lastId).style.display="none";

//revealing selected form

document.getElementById(id).style.display="block";
```

<u>Front End Theme and Style</u>

This file called the nmaStyle.css is charge of the dimensions of the website, how elements overflow, the color scheme, basically all color and physical appearances of the website are dictated here.

With a blank cascading style sheet the website would simply be white with black text with pages that would extend vertically infinitely since all the elements would be placed one after the other instead of overlayed the way in which they are style now.

In this respect I coded the website with a green theme since that is typically a clean, natural, and calming color.

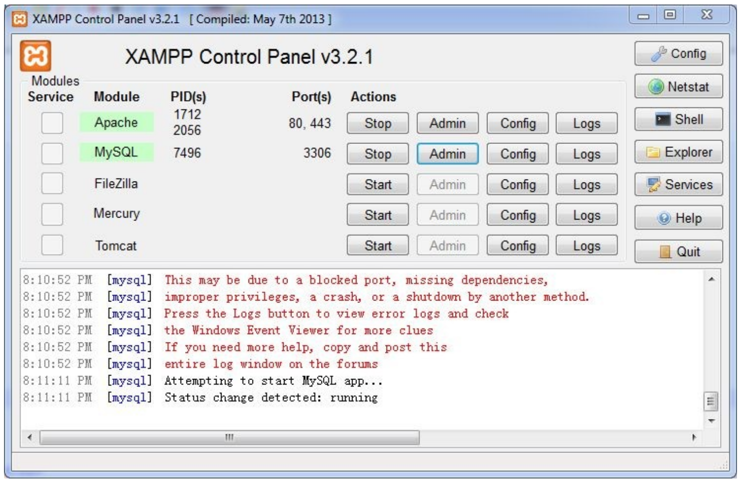
6. Creating an Application Instance and user Manual

The section describe how to create an instance of the application and is followed by a simple user manual.

1.) First the mysql database used should have a username and password set to ‘root’ and ‘toor’ respectively. This isn’t necessary if you wish you can also alter the information already in the account.php to match that of an existing mysql account

2.) Create a folder within the htdocs, (after having installed xampp), name it nma and place the contents of this folder into it. You can also simply drag the entire nma folder into htdocs.

3.) Create the tables and populate them using the sql statements and inserts provided in the sqlRelations&Data.doc file.



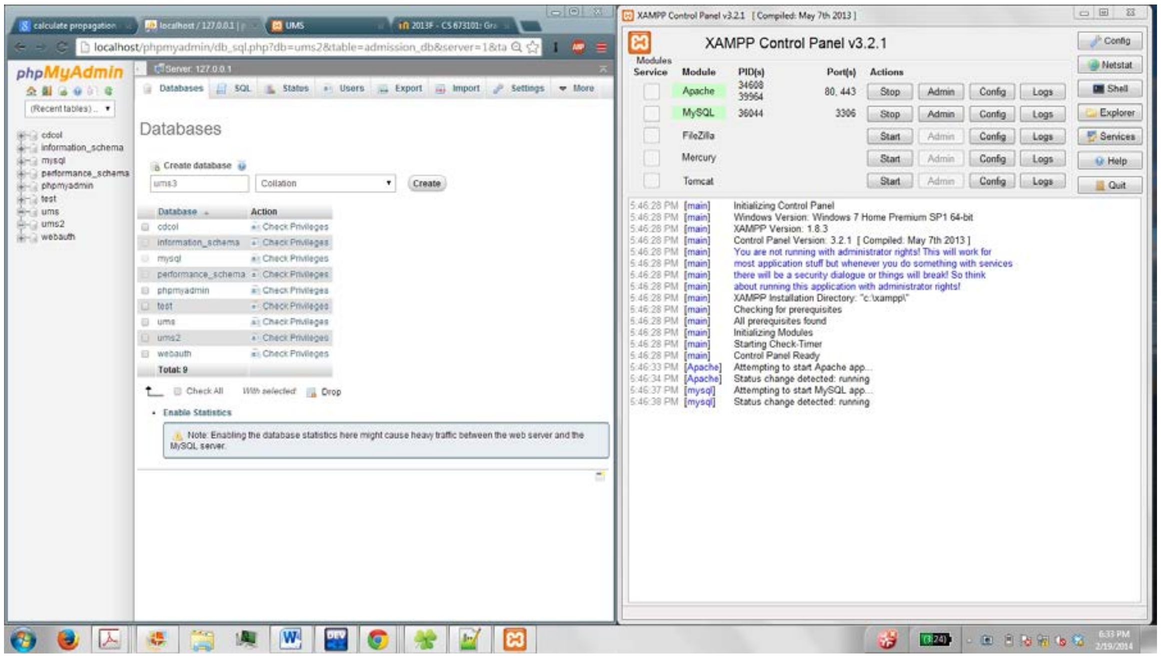
```
8:10:52 PM  [mysql]  This may be due to a blocked port, missing dependencies,
8:10:52 PM  [mysql]  improper privileges, a crash, or a shutdown by another method.
8:10:52 PM  [mysql]  Press the Logs button to view error logs and check
8:10:52 PM  [mysql]  the Windows Event Viewer for more clues
8:10:52 PM  [mysql]  If you need more help, copy and post this
8:10:52 PM  [mysql]  entire log window on the forums
8:11:11 PM  [mysql]  Attempting to start MySQL app...
8:11:11 PM  [mysql]  Status change detected: running
```

Step 1:

Start Apache and MySql

Visit localhost/phpmyadmin

Click Home, Databases, then Create a new DB (in this case NMA)



localhost/phpmyadmin/db_sql.php?db=ums2&table=admission_db&server=1&ta

Server: 127.0.0.1

Databases

Create database: ums3 | Collation | Create

Database	Action
cdcol	Check Privileges
information_schema	Check Privileges
mysql	Check Privileges
performance_schema	Check Privileges
phpmyadmin	Check Privileges
test	Check Privileges
ums	Check Privileges
ums2	Check Privileges
webauth	Check Privileges
Total: 9	

Note: Enabling the database statistics here might cause heavy traffic between the web server and the MySQL server.

XAMPP Control Panel v3.2.1 [Compiled: May 7th 2013]

Service	Module	PID(s)	Port(s)	Actions			
	Apache	34608 39964	80, 443	Stop	Admin	Config	Logs
	MySQL	36044	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

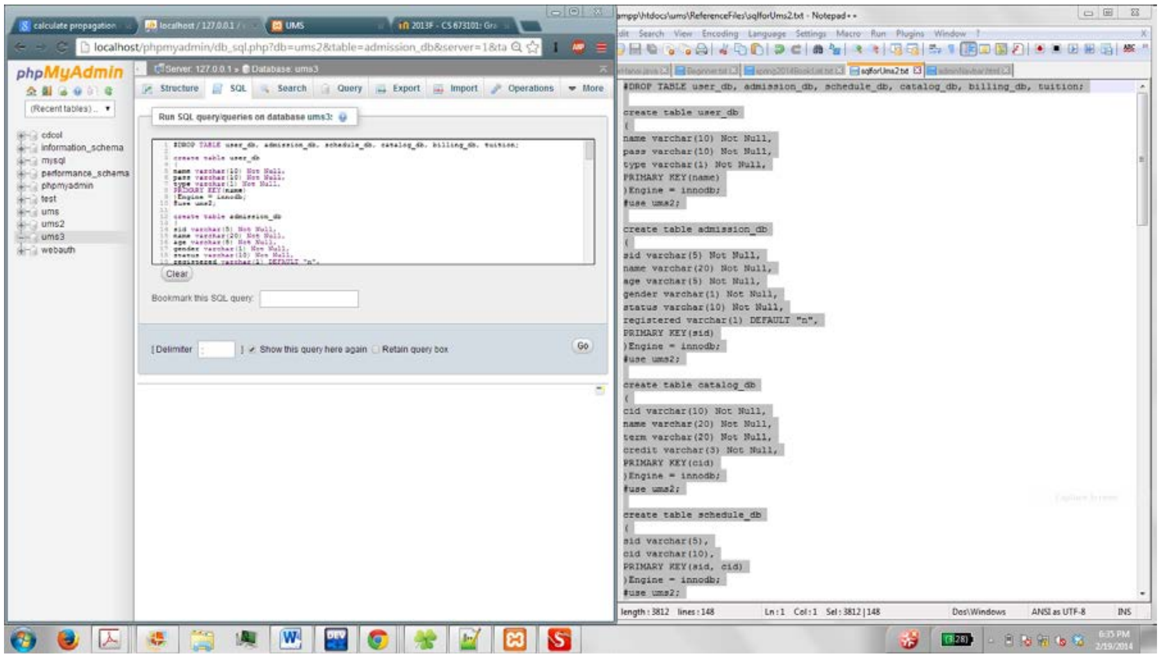
```
5:46:28 PM  [main]    Initializing Control Panel
5:46:28 PM  [main]    Windows Version: Windows 7 Home Premium SP1 64-bit
5:46:28 PM  [main]    XAMPP Version: 1.8.3
5:46:28 PM  [main]    Control Panel Version: 3.2.1  [ Compiled: May 7th 2013 ]
5:46:28 PM  [main]    You are not running with administrator rights! This will work for
5:46:28 PM  [main]    most application stuff but whenever you do something with services
5:46:28 PM  [main]    there will be a security dialogue or things will break! So think
5:46:28 PM  [main]    about running this application with administrator rights!
5:46:28 PM  [main]    XAMPP Installation Directory: "c:\xampp\"
5:46:28 PM  [main]    Checking for prerequisites
5:46:28 PM  [main]    All prerequisites found
5:46:28 PM  [main]    Initializing Modules
5:46:28 PM  [main]    Starting Check-Timer
5:46:28 PM  [main]    Control Panel Ready
5:46:33 PM  [Apache]  Attempting to start Apache app...
5:46:34 PM  [Apache]  Status change detected: running
5:46:37 PM  [mysql]   Attempting to start MySQL app...
5:46:38 PM  [mysql]   Status change detected: running
```

Step 2:

Click newly created Database

Press SQL

Then copy and paste the sql provided to create the tables, (there is sample inputs included)

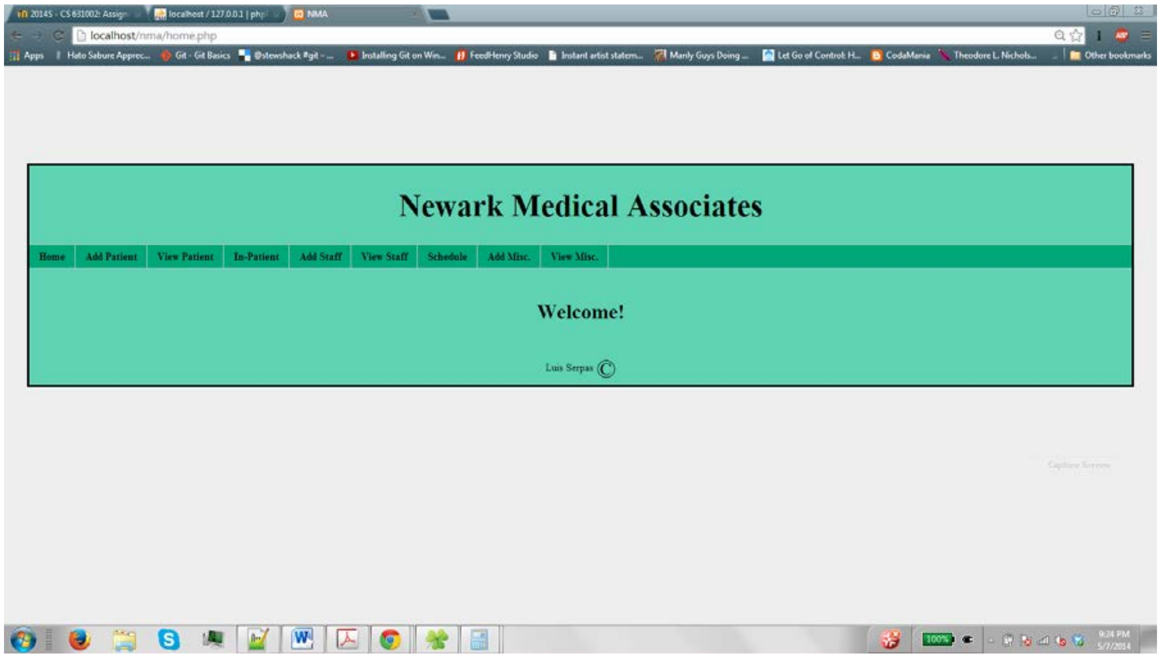


```
#DROP TABLE user_db, admission_db, schedule_db, catalog_db, billing_db, tuition;

create table user_db
(
name varchar(10) Not Null,
pass varchar(10) Not Null,
type varchar(1) Not Null,
PRIMARY KEY(name)
)Engine = innodb;
#use ums2;

create table admission_db
(
sid varchar(5) Not Null,
name varchar(20) Not Null,
age varchar(5) Not Null,
gender varchar(1) Not Null,
status varchar(10) Not Null,
registered varchar(1) DEFAULT "n",
PRIMARY KEY(sid)
)Engine = innodb;
#use ums2;

create table catalog_db
(
cid varchar(10) Not Null,
name varchar(20) Not Null,
term varchar(20) Not Null,
credit varchar(3) Not Null,
PRIMARY KEY(cid)
)Engine = innodb;
#use ums2;

create table schedule_db
(
sid varchar(5),
cid varchar(10),
PRIMARY KEY(sid, cid)
)Engine = innodb;
#use ums2;
```

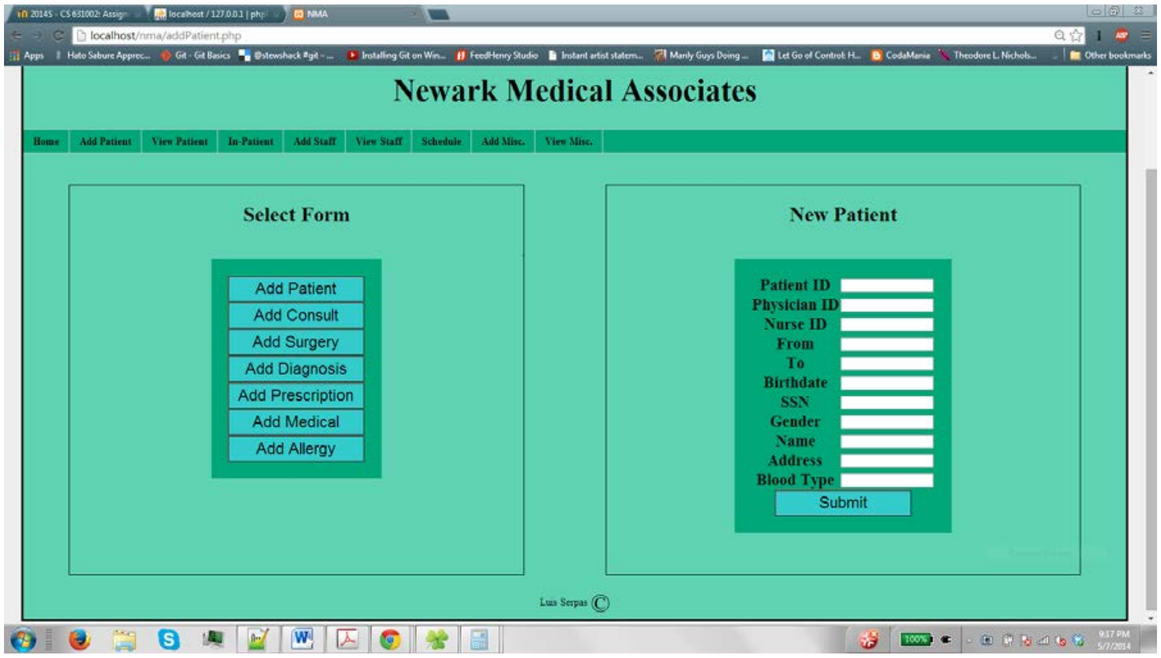
Step 3:

After transferring content into NMA in htdocs

Visit localhost/NMA /home.php to access the home page.



Step 4:



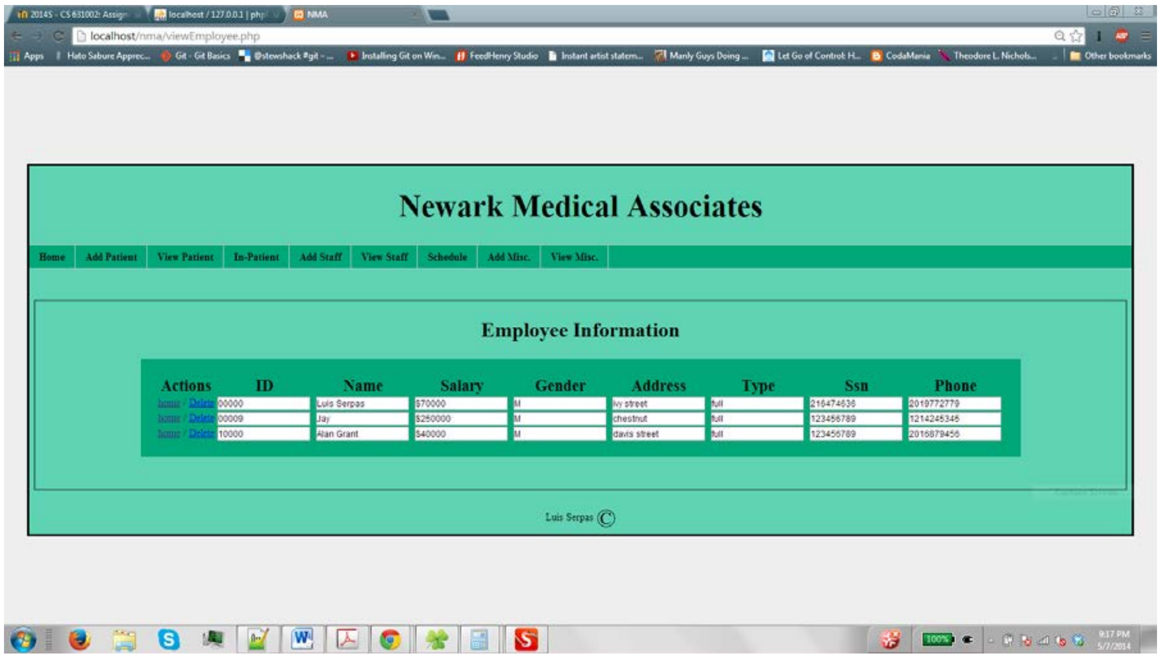
This is screen shot of the main user interface provided by the application.

On the left there is a list of buttons that upon clicked will reveal the related form. The user can then insert the necessary data and submit it the php files which will in turn insert it into the sql database.

Step 5:

The following is the next basic functionality provided by the website. All information from a table is provided for the user by populating a table located within the html page. To access any one of this table a user simply clicks a button akin to those shown in the previous image. Upon doing so the user will be sent to a page similar to the own displayed below.

Certain tables provide the user with the ability to update the data in the table. Such functionality is evident through the provided submit button that isn't absent in most read only tables.



Newark Medical Associates

Employee Information

Actions	ID	Name	Salary	Gender	Address	Type	Ssn	Phone
home / Delete	00000	Luis Serpas	$70000	M	ivy street	full	216474636	2019772779
home / Delete	00009	Jay	$250000	M	chestnut	full	123456789	1214245345
home / Delete	10000	Alan Grant	$40000	M	davis street	full	123456789	2016879456

Luis Serpas ©